モバイルクラウド環境における属性ベース暗号の改良

石黒 司　　清本 晋作　　三宅 優

株式会社 KDDI 研究所
356-8502 埼玉県ふじみ野市大原 2-1-15
{tsukasa, kiyomoto, miyake}@kddilabs.jp

あらまし 属性ベース暗号は安全なアクセス制御を実現する方式として研究されているが, 暗号化や復号にかかる時間が復号を許可する属性数のオーダとなるため, 処理能力が低いモバイル端末などでは計算に時間を要することがあった. 本稿ではモバイルクラウド環境に適した属性ベース暗号方式を提案する. 本提案方式では, クラウド環境側でユーザ端末側の秘密を保持しつつ処理の一部を負担させることにより, ユーザ側のモバイル端末における計算量を $O(1)$ とすることができる. また, 不正者利用者や端末紛失者の鍵を失効することも可能となっている. 本稿で提案する方式の利用により, クラウド環境においてサーバ側の不正にも安全性を保つことが可能であり, 特に計算資源の乏しいモバイル端末で属性ベース暗号によるアクセス制御を実現することが可能となる.

Improvement of Attribute-based Encryption in Mobile Cloud Computing

Tsukasa Ishiguro　　Shinsaku Kiyomoto　　Yutaka Miyake

KDDI R&D Laboratories, Inc.
2-1-15 Ohara, Fujimino-shi, Saitama, 356-8502 Japan
{tsukasa, kiyomoto, miyake}@kddilabs.jp

Abstract Attribute-based encryption(ABE) realises a cryptographic access control based on user attributes. However, one of the main efficiency drawbacks of ABE is that time required to encryption and decryption grows with the complexity of the access policy. In this paper, we propose a new ABE scheme suited for mobile cloud computing. Our scheme requires constant time to compute encryption and decryption in mobile devices. Additionally, we achieve key revocation and attribute hiding. Our solution enables lightweight devices to securely access based on user attributes with minimal cost even if the cloud server misbehaves.

1 はじめに

1.1 背景

近年, 新たな通信サービスとしてクラウドコンピューティングが広がりをみせている. クラウドコンピューティングの定義はいくつかあるが, 本稿では「ネットワークを通じて, 情報処理サービスを, 必要に応じて提供／利用する形の情報処理の仕組み」とする [21]. このクラウドコンピューティングによって, 利用者はサーバやネットワーク等の情報処理基盤を持つことなくサービスを享受することが可能となり, コストの削減, クラウド基盤の冗長化によるサービスの継続性, 柔軟なコンピュータリソースの追加と解放などの利点が生まれる. 一方, クラウド事業者から見るとアプリケーションの標準化や, 運用保守の効率化などがあり, 双方によって利点がある. 現在, 代表的なクラウドサービスとして Google 社の Google Apps[11] や, Amazon 社の Amazon Web Services[2], そして弊社の KDDI MULTI CLOUD[20] がある.

このように利用者・事業者にとってメリットがある一方で, セキュリティ上の懸念もある. 情報処理推進機 (IPA) によるとユーザ企業の懸念事項として, 「重要なデータを外部に預け, その管理状態が不透明になること」, 「顔の見えないクラウドベンダの内部者による犯行の可能性」, 「他の利用者など, 想定していなかった第三者（他のユーザ企業など）からのアクセスによる悪影響が生じてしまうこと」などが報告されている [16]. 同様の脅威が Cloud Security Alliance や ENISA においても指摘されている [8, 9]. そのため, クラウドサービス事業者や他のユーザ企業による不正を考慮する必要がある.

例えば, クラウドサービスの代表的な利用形態としてストレージサービス [1, 25] がある. このようなサービスを利用すると, 様々なデバイスや利用者でファイルを共有することが可能となる一方で, 利用者にとって重要な情報が外部に渡ってしまう危険性がある. そのため利用者の重要な情報資産を保護するためデータの暗号化やアクセス制御などを組み合わせてこの課題を解決する必要がある.

暗号の分野では属性ベース暗号によるアクセス制御が広く研究されている [27, 4, 15, 22, 7, 30, 29, 12, 13, 10, 26, 23]. 属性ベース暗号は利用者の属性情報 (“部長”, “GL”, “社員” 等) によって復号できる暗号文を規定する方式である. 属性ベース暗号によって暗号学的に安全なアクセス制御を実現することができ, 先に述べたような, クラウドサービス事業者や他のユーザ企業による不正なアクセスを防止することが可能となる. 属性ベース暗号には鍵規定型 (KP-ABE) と暗号文規定型 (CP-ABE) がある. KP-ABE では, ユーザに属性の規定 (“GL” かつ “案件 A 担当” かつ “案件 B 担当” 等) を持っており, 暗号文には属性 (“案件 C 担当” 等) が割り振られるているモデルである. CP-ABE は KP-ABE とは逆に, 暗号文に属性の規定が定められており, ユーザに属性に応じた鍵が割り振られているモデルである. また, 属性ベース暗号は復号規定の構造に関しても種類がある. 一つは木構造で規定するタイプであり, もう一つは AND-gate で規定するタイプである. 本稿では木構造で規定する CP-ABE を扱う.

属性ベース暗号ではペアリング関数の計算が支配的となる. ペアリング関数の高速実装はこれまで盛んに研究されてきた [5, 28, 3, 24]. これらの研究の結果, PC 上では高々数 msec の時間で 1 回のペアリング計算が可能となっている. 一方, 携帯電話上でのペアリング計算の実装結果もいくつか報告されている [18, 17, 19]. しかしながら, 処理能力 (CPU のクロック周波数, アーキテクチャ, メモリ量等) や OS の違いから, PC 上での実装と比べるとコストが数百倍程度大きい. そのため, 現状ではモバイル端末においてはペアリング計算が必要のない方式の方が望ましい.

属性ベース暗号においてモバイル端末におけるペアリ

ング計算を省く改良プロトコルが提案されている [30, 14]. [30] では属性ベース暗号の処理を分割し, ユーザのモバイル端末ではペアリング関数の計算を省いている. その他の処理も $O(1)$ で暗号化・復号処理を行い, それ以外の属性ベース暗号の処理はサーバに負担させている. また, これによって平文情報がサーバに知られることはないことを示している. [14] では, 復号処理に Proxy サーバを介させることによって, 端末にダウンロードする前にほとんどの復号処理を行う. モバイル端末では簡単な ElGamal 暗号を復号するだけで平文を取り出すことができる. 暗号化にかかる時間は従来通り属性数のオーダである. どちらの場合も, モバイル端末においての計算量を削減する効果は大きいが, 鍵の失効や属性の秘匿など, クラウドサービスにおいて考慮すべき課題がある.

本稿ではモバイルクラウドサービスにおいて必要となる要件を整理し, それらを解決する属性ベース暗号のプロトコルを提案する.

1.2 想定するクラウドサービスとその要求条件の定義

本節では, 想定するクラウドサービスの要求条件について定義する. 本稿では以下のサービスを考える.

複数のユーザがモバイル端末を所持し, クラウドストレージとデータのやり取りを行う. あるユーザは, 1 つもしくは複数の属性情報を持っており, 暗号化する際には, その暗号化されたデータを復号できる属性 (復号規定と呼ぶ) を決めることができる. クラウドストレージには暗号化されたデータが保管されており, 復号規定を満たすユーザのみがデータを復号することができる. 以下に要求条件をまとめる.

要求 1 ユーザは属性情報を用いて復号規定を定めて暗号化することができ, 復号規定を満足する場合のみ復号することができる.

要求 2 暗号化処理, 復号処理においてユーザのモバイル端末における計算量は小さくなければならない. 具体的には $O(1)$ でなければならない.

要求 3 暗号文あるいは復号規定から, 平文, 属性に関する情報が漏れてはならない.

要求 4 端末を紛失した場合や不正な利用者がいた場合, その鍵を失効することができる.

クラウドストレージサービスを利用する場合, 利用者の属性によってアクセス制御を行う方式が有用であり, **要求 1** はそれを示している.

次に, 本稿ではユーザの利用する端末をモバイル端末とする. モバイル端末は PC とは異なり計算資源は大きく制限される. そのため, **要求 2** の通り計算量を $O(1)$ とすることが必要となる. ここで, **要求 2** はユーザのモバイル端末における計算量のみを要求するものであり, 暗号化処理あるいは復号処理全体の計算量を規定するものではないことに注意されたい.

要求 3 は秘匿性の定義である. 平文の情報が漏れないことはもちろんであるが, 属性に関する情報も漏れてはならない.

最後に, **要求 4** は鍵失効機能の定義である. 特にモバイル端末は端末自体を紛失しやすく, また端末に対する改変を行うことも考えられるため, 不正な鍵を失効する機能が必要不可欠である.

本稿では要件 1-4 を満たす属性ベース暗号を提案する.

2 記号定義

本稿で用いる記号を定義する.

定義 1 (ペアリング写像 e)**.** $\mathbb{G}_0, \mathbb{G}_1$ は素位数 p の乗法群, $g \in \mathbb{G}_0$ を生成元とする. ペアリング写像 $e : \mathbb{G}_0 \times \mathbb{G}_0 \to \mathbb{G}_1$ は以下の *2* つの性質を満たす.

- 双線形性*:* $\forall u, v \in \mathbb{G}_0, a, b \in \mathbb{Z}_p$ に対して $e(u^a, v^b) = e(u, v)^{ab}$ を満たす.
- 非退化性*:* $e(u, u) \neq 1$

定義 2 (ラグランジュ係数)**.** $i, x \in \mathbb{Z}_p, S$ を $\mathbb{Z}_p$ を元の集合とするとき, ラグランジュ係数

$$\Delta_{i,S}(x) = \prod_{i \in S, j \neq i} \frac{x-j}{i-j}$$

とする.

定義 3 (ハッシュ関数 $H(\cdot)$)**.** 衝突困難なハッシュ関数 $H : \{0,1\}^* \to \mathbb{G}_0$ とする.

定義 4 (アクセス木)**.** $\mathcal{T}$ をアクセス木とする. $\mathcal{T}$ は葉でないノードには閾値 *(threshold gate)* を持っており, 子の論理式を表している. num_x をノード x の子の数とし, k_x を閾値とすると, $0 < k_x \leq num_x$ となっている. k_x は必要な属性数を表している. 例えば $k_x = 1$ であれば *OR* を表し, $k_x = num_x$ であれば *AND* を表している. 葉は属性を一つ表しているとし, $k_x = 1$ とする.

以下にアクセス木に関する関数を定義する. $parent(x)$ を x の親ノードとする. $att(x)$ を, x が葉である時, x が示す属性を表す. 全てのノードには番号が振ってあり, あるノードの子は *1* から num_x までとすると, $index(x)$ は x の番号を表す. 番号はユニークに振られているとする. $sibs(x)$ は x と同親の兄弟ノードを表す. 木 $\mathcal{T}$ の全てのノード数を $|\mathcal{T}|$ と表す.

定義 5 (DBDH 仮定 [6])**.** g を $\mathbb{G}$ の生成元とし, a, b, c, $z \in \mathbb{Z}_p^*$ をそれぞれランダムに選択する. *DBDH* 仮定とは $[g, g^a, g^b, g^c, e(g,g)^{abc}]$ と $[g, g^a, g^b, g^c, e(g,g)^z]$ を識別するような多項式時間アルゴリズムは存在しないという仮定である.

3 関連研究

本章では関連研究として CP-ABE,tk-CP-ABE,PP-CP-ABE, Green 等の方式について説明する.

3.1 CP-ABE[4]

CP-ABE は 5 つのアルゴリズムから構成される.

Setup$(\lambda) \to PK, MK$ セキュリティパラメータ λ を用いて, 公開鍵 PK, マスター鍵 MK を出力する.

Encrypt$(PK, M, \mathbb{A}) \to CT$ 入力を公開鍵 PK, 平文 M, アクセス構造 $\mathbb{A}$ とし, $\mathbb{A}$ で許可された属性を持つユーザが復号可能な暗号文 CT を出力する.

KeyGeneration$(MK, S) \to SK$ 入力をマスター鍵 MK, 属性集合 S とし, 秘密鍵 SK を出力する.

Decrypt$(PK, CT, SK) \to M$ 入力を公開鍵 PK, 暗号文 CT, 秘密鍵 SK とし, SK が CT で定めるアクセス構造で許可されていれば平文 M を出力する.

Delegate$(SK, \widetilde{S}) \to \widetilde{SK}$ 秘密鍵 SK と属性集合 S の部分集合 $\widetilde{S} \subseteq S$ を入力し, 部分集合 $\widetilde{S}$ に対応する秘密鍵 $\widetilde{SK}$ を出力する.

CP-ABE は木構造を用いたアクセス構造を復号規定とすることにより属性ベース暗号を構成している. 具体的な構成方法は [4] を参照されたい.

3.2 tk-CP-ABE[15]

tk-CP-ABE は CP-ABE を改良したプロトコルであり, 鍵の不正利用の防止を実現している.

Setup$() \to PK, MK$ 暗黙的に定まっているセキュリティパラメータを用いて, 公開鍵 PK, マスター鍵 MK を出力する.

Encrypt$(PK, M, \mathbb{A}) \rightarrow CT$ 入力を公開鍵 PK, 平文 M, アクセス構造 $\mathbb{A}$ とし, $\mathbb{A}$ で許可された属性を持つユーザが復号可能な暗号文 CT を出力する.
KeyGeneration$(MK, PK, ID, S, b) \rightarrow SK$ 入力をマスター鍵 MK, 属性集合 S, ユーザの ID, ユーザの個人情報 b とし, 秘密鍵 SK を出力する.
GetToken$(ID, C) \rightarrow T$ ユーザの ID に対応する $\hat{D}$ と暗号文の一部 C を入力とし, トークン T を出力する.
Decrypt$(PK, CT, SK, T, ID, b) \rightarrow M$ 入力を公開鍵 PK, 暗号文 CT, 秘密鍵 SK, トークン T, ユーザの ID, ユーザの個人情報 b とし, SK が CT で定めるアクセス構造で許可されていれば平文 M を出力する.

tk-CP-ABE はトークンサーバを用いることにより, 復号する際にユーザを認証している. それによって鍵の不正利用の防止を実現している. 詳細なプロトコルは [15] を参照されたい.

3.3 PP-CP-ABE[30]

PP-CP-ABE は CP-ABE を改良し, ユーザのモバイル端末の計算量を $O(1)$ まで削減したプロトコルである. この方式では, 復号規定のアクセス木の根の部分に"DO" という属性を追加する. そしてユーザはこの"DO" 属性に対する秘匿処理を行う. この計算量は $O(1)$ である. 詳細なプロトコルは [30] を参照されたい.

3.4 Green et al.[14]

[14] は CP-ABE を改良した方式であり, 復号にかかる計算量を $O(1)$ とすることができる. この方式では, まず Transformation Key(TK) をそれぞれのユーザが秘密鍵とは別に持つ. 暗号文を復号する際, まず Proxy-server でこの TK を用いて属性ベース暗号の復号処理を行う. その結果, 単純な ElGamal 暗号のスタイルになるため, ユーザは属性数には依存せずに $O(1)$ で復号することができる.

3.5 課題

本節で説明した関連研究では, 1 章で示した要求事項を部分的にしか満たすことはできない. 例えば, CP-ABE では**要求 1** を満たすが, 他を満たすことはできない. tk-CP-ABE では**要求 1**, **要求 4** を満たすが他を満たさない. PP-CP-ABE では**要求 1**, **要求 2** を満たすが他を満たさない. Green et al. の方式では PP-CP-ABE と同様, **要求 1**, **要求 2** を満たすが他を満たさない. 本稿では要求事項を 4 つとも満足するプロトコルを提案する.

4 提案モデル

本章では提案モデルについて説明する.

4.1 アルゴリズムの定義

Setup$(\lambda) \rightarrow PK, MK$ セキュリティパラメータ λ を入力とし, 公開鍵 PK, マスター鍵 MK を出力する.
KeyGeneration$(MK, PK, ID, S, b) \rightarrow SK$ 入力をマスター鍵 MK, 公開鍵 PK, 個人 ID を ID, 属性集合 S, ユーザの個人情報 b とし, 秘密鍵 SK を出力する.
Encrypt$_U(PK, M, \mathbb{A}) \rightarrow CT_U$ 入力を公開鍵 PK, 平文 M, アクセス構造 $\mathbb{A}$ とし, 中間暗号文 CT_U を出力する.
Encrypt$_S(PK, \mathbb{A}, CT_U) \rightarrow CT$ 入力を公開鍵 PK, アクセス構造 $\mathbb{A}$, 中間暗号文 CT_U とし, $\mathbb{A}$ で許可された属性を持つユーザが復号可能な暗号文 CT を出力する.
GetToken$(ID, \hat{C}) \rightarrow T$ ユーザの ID と暗号文の一部 $\hat{C}$ を入力とし, トークン T を出力する.
Decrypt$_S(\widetilde{SK}, CT) \rightarrow CT_S$ 暗号文 CT, マスクされた秘密鍵 $\widetilde{SK}$ とし, $\widetilde{SK}$ が CT で定めるアクセス構造で許可されていれば中間暗号文 CT_S を出力する.
Decrypt$_U(b, CT_S, t, T) \rightarrow M$ ユーザの秘密情報 b, 暗号文 CT_S, トークン T, マスクの値 t とし, 平文 M を出力する.
GetMaskKey$(SK, T) \rightarrow (\widetilde{SK}, t)$ 入力を秘密鍵 SK, トークン T とし, マスクされた秘密鍵 $\widetilde{SK}$, マスク値 t を出力する.

4.2 安全性定義

[15] によると, 満たすべき安全性は暗号文識別不可能性 (Ciphertext indistinguishability), 複製不可能性 (Uncloneability), プライバシ保護 (Privacy preserving) である. 本方式では属性秘匿性についても証明を行う.

まず安全性ゲームを定義する.
System Setup: Challenger は Setup(l) を動かし PK を得る.
Phase 1: Adversary は以下のクエリを多項式回実行することができる

- 属性集合 S を持つ, どのユーザの秘密鍵, 個人情報を得る. Adversary は ID,S を指定できる
- 属性集合 S を持つ, ユーザの秘密鍵を得る. Adversary は ID,S,b を指定できる.
- どのユーザのどの暗号文に対するトークンでも入手できる. Adversary は $\hat{C}$,ID を指定できる

Choose Policy: Adversary は復号規定 W を選ぶ.
Challenge: Adversary は平文 $M_0, M_1 (M_0 \neq M_1)$ をランダムに選択し, Challenger はランダムに $\alpha \in \{0, 1\}$ を選び, M_α を復号規定 W の元で暗号化する. そして Challenger は C_α を Adversary に与える.
Phase 2: Adversary は Phase1 を繰り返す.
Guess: Adversary は $\alpha' \in \{0, 1\}$ を出力する. この時, $\alpha = \alpha'$ であれば Charenger の勝ち.

このゲームにおいて, Adversary の advantage を $ADV := Pr[\alpha = \alpha'] - \frac{1}{2}$ とする. このモデルを non-Selective ID model と呼ぶ.

定義 6 (暗号文識別不可能性)**.** どのような多項式時間の *Adversary* に対しても ADV が *negligible* である時, *ABE* が *Chosen Plaintext Attack(CPA)* に対して *Selective ID model* の元で安全であるという.

定義 7 (Uncloneability)**.** 個人識別子 ID, 秘密鍵 D, 個人情報 b とする. この時, D を得ると, トークンサーバにアクセスし, b を $O(|B|)$ で計算できるとき, *ABE* は *Uncloneability* を満たすという. ここで $|B|$ はユーザの個人情報の *domain* である. $|B|$ は ℓ*(security parameter)* の *polynomial* で表される.

定義 8 (Strong Uncloneability)**.** 公開鍵を PK とする. $D' \subseteq D$ とする. C を暗号文, W を復号規定とする. ある多項式時間アルゴリズム $\mathcal{F}$ があって, *GetToken* オラクル, *Decrypt*$_S$ オラクルにアクセスできるとする. この時,

$$\mathcal{F}^{\{GetTolen(\cdot,\cdot),\mathrm{Decrypt}_S(\cdot,\cdot)\}} = \mathrm{Decrypt}_U(b, \mathrm{Decrypt}_S(\widetilde{SK}, CT), t, GetToken(ID, \hat{C}))$$

を用いて *extractor algorithm* χ が $\widetilde{SK}, CT, ID, t, \hat{C}$ からユーザ ID の個人情報 b を $O(|B|)$ で計算できるとき, そのアルゴリズムは *Strong Uncloneability* を満たすという.

定義 9 (Privacy preserving)**.** 公開鍵を PK, マスター鍵を MK とする. ユーザの個人識別子を ID, 秘密鍵を SK, トークン情報を $\hat{D}$, 属性集合を S, 個人情報を b とすると

$$\Pr[b|PK, MK, ID, S, \hat{D}] = \Pr[b|ID, S]$$

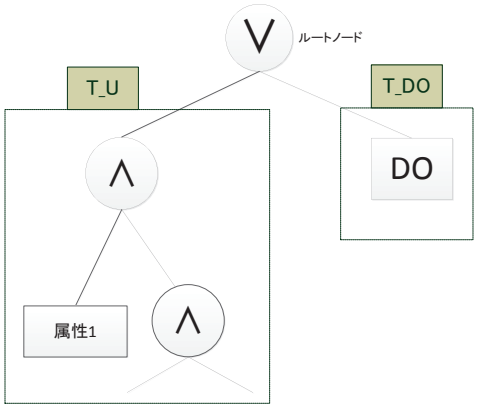

Figure 1: アクセス木

のとき, この *ABE* スキームは *Privcacy Preserving* であるという.

最後に, 属性識別不可能性を定義する. そのために, 上述の安全性ゲームの Challenge フェーズと Guess フェーズを以下のように書き換える.

Challenge: Adversary は平文 M をランダムに選択し, 復号規定 V_0, V_1 をランダムに選択する. Challenger はランダムに $\beta \in \{0,1\}$ を選び, M を復号規定 V_β の元で暗号化し, C_β として Adversary に与える.

Guess: Adversary は $\beta' \in \{0,1\}$ を出力する. この時, $\beta = \beta'$ であれば Charenger の勝ち.

このゲームにおいて, Adversary の advantage を $ADV := Pr[\beta = \beta'] - \frac{1}{2}$ とする.

定義 10 (属性識別不可能性)**.** どのような多項式時間の *Adversary* に対しても, $ADV := Pr[\beta = \beta'] - 1/2$ が *negligible* であるとき, *ABE* が *CPA* に対して *non-Selective ID model* の元で属性識別不可能性を持つという.

5 提案方式

本方式は tk-CP-ABE 方式のトークンサーバを用いた端末の認証機能と, PP-CP-ABE 方式のモバイル端末における計算量削減手法を組み合わせて実現する. 暗号化する際にはユーザが復号規定 $\mathcal{T}_\mathrm{U}$ を選ぶ. そして, ユーザはダミーの属性情報"*DO*" を設定し, "*DO*" 属性一つからなる部分木 $\mathcal{T}_\mathrm{DO}$ を構成する (図 1). 暗号化全体では $\mathcal{T}_\mathrm{U} \wedge \mathcal{T}_\mathrm{U}$ について行うが, ユーザは *DO* と $\mathcal{T}_\mathrm{U}$ のルートノードからなる木に対してのみ計算を行う. そのため, ユーザの計算量は $O(1)$ となる. $\mathcal{T}_\mathrm{U}$ のさらに下の部分木に関してはサーバ側で残りの計算を行う.

また, ユーザは復号する際, 必ずトークンサーバにアクセスする必要がある. そのため, トークンサーバに保存された ID に対応する値を削除することにより鍵の失効を行うことが出来る.

PP-CP-ABE 方式では, 復号規定をそのままクラウド側の関数である $\mathrm{Encrypt}_S$ に渡していたが, そのままでは属性値に関する情報がクラウド側に漏れてしまう. そのため, $\mathrm{Encrypt}_U$ では乱数でべき乗計算を行うことにより値の性質を保ったまま属性値を秘匿している. この処理の追加のため, $\mathrm{Encrypt}_U$ の計算量が復号規定に定められた属性数に依存してしまうが, 事前計算を利用することによって $O(1)$ に抑えることが出来る (詳細は 6.2 節参照).

以下に, それぞれのアルゴリズムの処理を示す.

Setup$(\lambda) \rightarrow PK, MK$　生成元 $g \in \mathbb{G}_0$ をランダムに生成し, $\alpha, \beta \in \mathbb{Z}_p$ をランダムに選択する. $PK = (g, e(g,g)^\alpha), MK = (g^\alpha)$ として出力する.

KeyGeneration$(MK, PK, ID, S, b) \rightarrow SK$　$r, b \in \mathbb{Z}_p$ をランダムに選択する. $\forall j \in S, r_j$ をランダムに計算する. SK を $SK = (b, \forall j \in S : D_j = g^r \cdot H(j)^{r_j}, D'_j = g^{r_j})$ として計算する. 最後に $\hat{D} = g^{\frac{(\alpha+r)}{b}}$ を計算し, トークンサーバに $(ID, \hat{D})$ を送る.

Encrypt$_U(PK, M, \mathcal{T}_\mathrm{U}) \rightarrow CT_U$　1 次式 q_R をランダムに選択する. $s = q_R(0), s_1 = q_R(1), s_2 = q_R(2)$ を計算する. アクセス木 $\mathcal{T}_\mathrm{U}$ の属性値 $att(y)$ を $H'_y = (H(att(y))^s$ で置き換え, $\mathcal{T}_\mathrm{S}$ とする. つまり, $\{\forall y \in Y_\mathrm{U} : H'_y = (H(att(y))^s,\ g'_y = g^s\}$ を計算する. そして以下のように CT_U を求める.

$$\begin{aligned} CT_U \quad = \quad & (s_1, \mathcal{T}_\mathrm{S}, \hat{C} = g^{s^2}, \widetilde{C} = M \cdot e(g,g)^{\alpha s}; C = h^s; \\ & \forall y \in Y_{DO} : C_y = g^{s q_y(0)}, C'_y = H(att(y))^{s q_y(0)};) \end{aligned}$$

Encrypt$_S(PK, \mathcal{T}_\mathrm{S}, CT_U) \rightarrow CT$　$\mathrm{Encrypt}(s_1, \mathcal{T}_\mathrm{S})$ を計算し, $CT_S = (\forall y \in Y_\mathrm{S} : C_y = g_y^{'q_y(0)}, C'_y = H_y^{'q_y(0)})$ を求める. CT を $CT = (\mathcal{T} = \mathcal{T}_\mathrm{S} \wedge \mathcal{T}_\mathrm{DO}; \widetilde{C}; C; \hat{C}; \forall y \in Y_\mathrm{S} \cup Y_\mathrm{DO} : C_y = g_y^{'q_y(0)}, C'_y = H_y^{'q_y(0)})$ として計算する.

GetToken$(ID, \hat{C}) \rightarrow T$　ID をデータベースから照合し, $T = e(\hat{C}, \hat{D}) = e(g^{s^2}, g^{\frac{\alpha+r}{b}}) = e(g,g)^{\frac{s^2(\alpha+r)}{b}}$ を出力する.

GetMaskKey$(SK, T) \rightarrow (\widetilde{SK}, t)$　ランダムに $t \in \mathbb{Z}_p$ を選択する. $\widetilde{\mathrm{T}} = T^t = e(g,g)^{\frac{ts(\alpha+r)}{b}}$, $\widetilde{SK} = (\widetilde{T}, \forall j \in S : \mathrm{D}_j = g^r \cdot H(j)^{r_j}, D'_j = g^{r_j})$ とする.

Decrypt$_S(\widetilde{SK}, CT) \rightarrow CT_S$　R を $Y_\mathrm{S} \cup Y_\mathrm{DO}$ のルートノードとし, $\mathrm{DecryptNode}(\widetilde{SK}, CT, R) \rightarrow A = e(g,g)^{rs^2}$ を計算する. $CT_S = (A, \widetilde{C})$ とする.

Decrypt$_U(b, CT_S, t, T) \rightarrow M$　$T' = T^{\frac{b}{t}}$ を計算する. 以下の計算により M を求める.

$$\begin{aligned} \frac{\widetilde{C}A}{T'} \quad &= \quad \frac{Me(g,g)^{\alpha s} \cdot e(g,g)^{rs^s}}{(e(g,g)^{\frac{ts^2(\alpha+r)}{b}})^{\frac{b}{t}}} \\ &= \quad \frac{Me(g,g)^{(\alpha+r)s^2}}{e(g,g)^{(\alpha+r)s^2}} \\ &= \quad M \end{aligned}$$

Subroutin;Encrypt$(s, \mathcal{T}) \rightarrow CT$　木 $\mathcal{T}$ の全てのノード x に対する多項式 q_x を選択する. ルートノードを R として, 次数をそれぞれ $d_x = k_x - 1$ とする. まず root R に対して, $q_R(0) = s \in \mathbb{Z}_p$ とする. R のその他の係数をランダムに決定する. R 以外のノード x に対して $q_x(0) = q_{parent(x)}(index(x))$ とし, その他の係数をランダムに決定する. Y を $\mathcal{T}$ の葉ノードの集合とし, 以下のように CT を計算する.

$$CT \quad = \quad (\forall y \in Y : C_y = g^{q_y(0)}, C'_y = H(att(y))^{q_y(0)})$$

Table 1: 計算量

	Hash	Mul $\mathbb{G}_0$	Exp $\mathbb{G}_0$	Mul $\mathbb{G}_1$	Inv $\mathbb{G}_1$	Exp $\mathbb{G}_1$	Pairing
Setup	-	-	1	-	-	-	1
KeyGeneration	$(\|S\|)^*$	$\|S\|$	$2\|S\|+2$	-	-	-	-
Encrypt$_U$	$(\|\mathcal{T}\|+1)^*$	-	$4+(\|\mathcal{T}\|+1)^*$	1	-	1	-
Encrypt$_S$	$(\|\mathcal{T}\|)^*$	-	$2\|\mathcal{T}\|$	-	-	-	-
Decrypt$_U$	-	-	-	2	1	1	-
Decrypt$_S$	-	-	-	$2\|\mathcal{T}\|$	$\|\mathcal{T}\|$	$\|\mathcal{T}\|$	$2\|\mathcal{T}\|+2$
GetToken	-	-	-	-	-	-	1
GetMaskKey	-	-	-	-	-	1	-

*事前計算可能, S：ユーザに付与する属性集合, $|\mathcal{T}|$：復号を許可する復号規定の葉の数

Subroutine:DecryptNode(SK, CT, R) $\rightarrow e(g,g)^{rs^2}$
全ての $\ell \in \mathcal{T}$ について
$\rho(\ell) = (\ell, \mathrm{parent}(\ell), \mathrm{parent}(\mathrm{parent}(\ell)), \cdots, R)$
とする. 次に, 全ての ℓ について

$$z_\ell = \prod_{x\in\rho(\ell), x\neq R} \Delta_{i,S}(0), \text{where } i=\mathrm{index}(x),$$
$$S=\{\mathrm{index}(y) \mid y\in \mathrm{sibs}(x)\}$$
$$= \prod_{x\in\rho(\ell), x\neq R} \prod_{j\in S, j\neq i} \frac{-j}{i-j}$$

を求める. 最後に

$$\mathrm{DecryptNode}(SK, CT, R) = \prod_{\ell\in L, i=\mathrm{att}(\ell)} \left(\frac{e(D_i, C_\ell)}{e(D'_i, C'_\ell)}\right)^{z_\ell}$$
$$= e(g,g)^{rs^2}$$

を出力する.

6 評価

本章では提案方式の評価を行う. 最初に安全性の証明を行う. 次に, 提案方式の効率性 (計算量) を評価し, 最後に, 1 章で示した要求条件を満足することを示す.

6.1 安全性の証明

本節では安全性の証明を与える. 具体的には暗号文識別不可能性, Uncloneability, プライバシ保護, 属性秘匿性について行う.

定理 1. 提案方式は選択平文攻撃に対して *non-selective-ID* モデルで *DBDH* 仮定の下で暗号文識別不可能性を満たす.

証明 1. *[15]* と同様な手法で証明することができるが紙面の都合上割愛する. □

定理 2. 提案方式は *Strong Uncloneability* を満たす.

証明 2. $\mathcal{F}$ を用いて χ が b を $O(|B|)$ で求めることを示す. χ への入力は PK, ID, C, D' である.
まず, $\mathcal{F}$ を用いて, M を求める. 次に, χ は PK,C,D' から, $\widetilde{A} = e(g,g)^{rs^2}$ を求める. ここで, $M = \frac{\widetilde{C}\widetilde{A}}{T^{\frac{b}{r}}}$ となるので, 全数探索することにより $O(|B|)$ でこの式を満たす b を見つけることができる. 従って提案方式は *Uncloneability* を満たす. □

定理 3. 提案方式は *Privacy preserving* である.

証明 3 (定理 3)**.** 提案方式において, PK, MK は b に依存せず選ばれているので,

$$\Pr[b|PK, MK, ID, S, \hat{D}] = \Pr[b|ID, S, \hat{D}]$$

が成り立つ. また, $\hat{D} = g^{\frac{(\alpha+r)}{b}}$ であり, G の位数は q であるので,

$$\Pr[b|ID, S, \hat{D}] = \Pr[b|ID, S, \frac{(\alpha+r)}{b} \bmod p]$$

ここで, α, r, b は独立なので,

$$\Pr[b|ID, S, \frac{(\alpha+r)}{b} \bmod p] = \Pr[b|ID, S]$$

を満たす. 従って提案方式は *Privacy preserving* を満たす. □

定理 4. 提案方式は, *CPA* に対して *non-Selective ID model* において暗号文識別不可能性を満たすとき, 属性識別不可能性を満たす.

証明 4. $M = M_0$, $V_0 = W, C_\beta = C_\alpha$ とし属性識別不可能性を破る *Adversary* を動作させ, 出力 β' を得る. ここで, V_1 はランダムな復号規定を選ぶ. $\beta' = 0$ であるとき, $\alpha' = 0$ を出力する. それ以外では, 確率 1/2 でランダムに選択した α' を出力する. その結果, 無視できない確率で暗号文識別不可能性を破ることができるため, 暗号文識別不可能性に反することになる. 以上より, 属性識別不可能性を満たす. □

6.2 効率性

本章では提案方式の計算量について議論する. 提案方式では特にユーザの計算量 (具体的には Encrypt$_U$, Decrypt$_U$) の削減が目的である.
復号する際, ユーザは Decrypt$_U$ の前に GetRandomKey を計算するが, この計算は $O(1)$ である. 同様に Decrypt$_U$ の計算量も $O(1)$ となるため, 復号全体でユーザの計算量は $O(1)$ である.
Encrypt$_U$ の計算量は, 復号規定のリーフノードの数を $|\mathcal{T}|$ とすると, $O(|\mathcal{T}|)$ となる. これは復号規定で選択した属性全てに対して $H'_y = (att(y))^s$ を計算するためである. しかし, この計算は M には依存せずに行うことができる. そのため, s, s_1, s_2 と全ての属性に対する H'_y を事前計算して保持しておくことにより, 毎回の暗号化計算から省略することができる. 従って Encrypt$_U$ の計算量は $O(1)$ となる.
全体としての計算量は暗号化・復号ともに $O(|\mathcal{T}|)$ である. ワーストケースとしてはシステム全体の属性数を N とすると $O(N)$ となる. この計算量はこれまでの ABE の計算量と同様である.
提案方式の全てのアルゴリズムの計算量を表 1 に示す.

6.3 要求事項の検証

本節では提案プロトコルと関連研究における要求事項の検証を行う. 関連研究は 3 章で説明した CP-ABE,tk-CP-ABE,PP-CP-ABE,Green 等の方式とする. その 4 つのプロトコルが要求事項を満足するかどうかについては 3.4 節に示した通りである.
要求 1 については, 属性ベース暗号の性質から満足することは自明である. また 6.2 節で示した事前計算手法を利用することにより暗号化計算・復号計算の計算量を $O(1)$ とすることができるため, **要求 2** についても満足する. **要求 3** については, 6.1 節に示した通りサーバ側の結託攻撃に対して平文・属性値に関する情報は洩れないため, 満足することができる. 最後に, 本方式はトークンサーバを

用いることにより, 不正利用者・端末紛紛失者の鍵を失効することが可能であり, **要求 4** を満足することができる. 従って, 本提案方式は要求 1-4 を全て満足する. 要求事項の比較を表 2 に示す.

Table 2: 要求事項の検証

	要求 1	要求 2	要求 3	要求 4
CP-ABE[4]	○	×	×	×
tk-CP-ABE[15]	○	×	×	○
PP-CP-ABE[30]	○	○	×	×
Green et al.[14]	○	△ *	×	×
This work	○	○	○	○

* 復号計算のみ $O(1)$

7 おわりに

属性ベース暗号は, クラウドコンピューティングにおいて安全なアクセス制御技術として適している. しかし, 属性数に比例して計算量が増大してしまうため, モバイル端末で暗号化・復号を行うことは困難であった. そこで, 本方式を適用することによってユーザのモバイル端末での計算量が $O(1)$ となり, どのようなスケールのシステムであっても一定の計算量で行うことが可能である. また, 不正者利用者の鍵の失効が可能となり, 属性に関する情報の秘匿も可能となった. 本方式を用いることにより, クラウドサーバ自身の安全性に依らずにシステムの安全性を高めることが可能となり, また, モバイルを利用したクラウドサービスにおいても高い効率性を持つことができるようになった.

References

[1] Amazon. Amazon S3. `http://aws.amazon.com/jp/s3/`.

[2] Amazon. Amazon Web Services. `http://aws.amazon.com/jp/`.

[3] D. Aranha, J. López, and D. Hankerson. High-speed parallel software implementation of the η_t pairing. In *Topics in Cryptology - CT-RSA 2010*, Vol. 5985 of *Lecture Notes in Computer Science*, pp. 89–105. Springer, 2010.

[4] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. *Security and Privacy, IEEE Symposium on*, Vol. 0, pp. 321–334, 2007.

[5] J. Beuchat, J. E. G. Díaz, S. Mitsunari, E. Okamoto, F. Rodríguez-Henríquez, and T. Teruya. High-speed software implementation of the optimal ate pairing over barreto-naehrig curves. Cryptology ePrint Archive, Report 2010/354, 2010.

[6] D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. Advances in Cryptology — CRYPTO 2001, 第 2139 巻 of *Lecture Notes in Computer Science*, pp. 213–229. Springer, 2001.

[7] M. Chase. Multi-authority attribute based encryption. In *Theory of Cryptography*, Vol. 4392 of *LNCS*, pp. 515–534. Springer, 2007.

[8] Cloud Security Alliance. Security guidance for critical areas of focus in cloud computing, 2009. `http://tinyurl.com//ycrchqj`.

[9] Cloud Security Alliance. Top threats to cloud computing, 2010. `http://tinyurl.com/yer9tvs`.

[10] C. Gentry and A. Silverberg. Hierarchical id-based cryptography. Advances in Cryptology — ASIACRYPT 2002, 第 2501 巻 of *Lecture Notes in Computer Science*, pp. 149–155. Springer, 2002.

[11] Google. Google App for Buiziness. `http://www.google.com/apps/intl/ja/business/index.html`.

[12] V. Goyal, A. Jain, O. Pandey, and A. Sahai. Bounded ciphertext policy attribute based encryption. In *Automata, Languages and Programming*, Vol. 5126 of *Lecture Notes in Computer Science*, pp. 579–591. Springer, 2008.

[13] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM conference on Computer and communications security*, CCS '06, pp. 89–98. Algorithms and Computation in Mathematics, 2006.

[14] M. Green, S. Hohenberger, and B. Waters. Outsourcing the decryption of abe ciphertexts. In *Usenix Security 2011*. 2011.

[15] M. J. Hinek, S. Jiang, R. Safavi-Naini, and S. F. Shahandashti. Attribute-based encryption with key cloning protection. Cryptology ePrint Archive, Report 2008/478, 2008.

[16] 情報処理推進機構 (IPA). クラウドコンピューティング社会の基盤に関する研究会報告書. `http://www.ipa.go.jp/about/research/2009cloud/pdf/100924_cloud.pdf`.

[17] 井山政志, 清本晋作, 福島和英, 田中俊昭, 高木剛. Android 携帯電話におけるペアリング暗号の実装. コンピュータセキュリティシンポジウム (CSS2010), 2B1-3. 情報処理学会, 2010.

[18] T. Iyama, S. Kiyomoto, K. Fukushima, T. Tanaka, and T. Takagi. Efficient implementation of pairing on brew mobile phones. In *Advances in Information and Computer Security*, Vol. 6434 of *Lecture Notes in Computer Science*, pp. 326–336. Springer, 2010.

[19] Y. Kawahara, T. Takagi, and E. Okamoto. Efficient implementation of tate pairing on a mobile phone using java. In *Computational Intelligence and Security*, Vol. 4456 of *Lecture Notes in Computer Science*, pp. 396–405. Springer, 2007.

[20] KDDI. KDDI MULTI CLOUD. `http://www.kddi.com/business/pr/multicloud/index.html`.

[21] 経済産業省商務情報政策局.「クラウドコンピューティングと日本の競争力に関する研究会 報告書」, 2010. `http://www.meti.go.jp/press/20100816001/20100816001-3.pdf`.

[22] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In *Advances in Cryptology - EUROCRYPT 2010*, Vol. 6110 of *Lecture Notes in Computer Science*, pp. 62–91. Springer, 2010.

[23] A. Lewko and B. Waters. Unbounded hibe and attribute-based encryption. Cryptology ePrint Archive, Report 2011/049, 2011.

[24] M. Naehrig, R. Niederhagen, and P. Schwabe. New software speed records for cryptographic pairings. Cryptology ePrint Archive, Report 2010/186, 2010.

[25] Nifty. Nifty Cloud. `http://cloud.nifty.com/`.

[26] R. Ostrovsky, A. Sahai, and B. Waters. Attribute-based encryption with non-monotonic access structures. Cryptology ePrint Archive, Report 2007/323, 2007.

[27] A. Sahai and B. Waters. Fuzzy identity-based encryption. In *Advances in Cryptology - EUROCRYPT 2005*, Vol. 3494 of *Lecture Notes in Computer Science*, pp. 557–557. Springer, 2005.

[28] M. Scott. On the efficient implementation of pairing-based protocols. Cryptology ePrint Archive, Report 2011/334, 2011.

[29] B. Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In *Public Key Cryptography - PKC 2011*, Vol. 6571 of *Lecture Notes in Computer Science*, pp. 53–70. Springer, 2011.

[30] Z. Zhou and D. Huang. Efficient and secure data storage operations for mobile cloud computing. Cryptology ePrint Archive, Report 2011/185, 2011.